Electricity plan recommender system with electrical instruction-based recovery

Junjie Zheng[a], Chun Sing Lai[a,b*], Haoliang Yuan[a], Zhao Yang Dong[c], Ke Meng[c], Loi Lei Lai[a]
a. School of Automation, Guangdong University of Technology Guangzhou, China
b. Department of Electronic and Computer Engineering, Brunel University London, London, UK
c. Department of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia
* Corresponding author: chunsing.lai@brunel.ac.uk
First and second authors contribute equally.

Abstract：

Several electricity tariffs have emerged for Demand Side Management (DSM), and residential customers are faced with challenges to choose the plan satisfying their personal needs. Electricity Plan Recommender System (EPRS) can alleviate the problem. This paper proposes a novel EPRS model named EPRS with Electrical Instruction-based Recovery (EPRS-EI), which is a dual-stage model consisting of feature formulation stage and recommender stage. In feature formulation stage, matrix recovery with electrical instructions is applied to recover appliance usages, and the recovered data is set as features representing customers' living patterns. In the recommender stage, Collaborative Filtering Recommender System (CFRS) based on K-Nearest Neighbors (KNN) and adjusted similarity is applied to recommend personal electricity plans to customers based on the above features. Different from other EPRS models, EPRS-EI is the first model utilizing matrix recovery methods and similarity computation with electrical instructions. With these electrical instructions, the proposed model is possible to utilize more explicit features and recommended more personalized plans. We then apply EPRS-EI to predict the testing customers' preference for electricity plans. Simulation results on recovering electricity data and their applications in EPRS confirm the effectiveness of our proposed methods in comparison to state-of-the-art methods, with 93.56%-94.85% customers correctly recommended.

Keywords:
Matrix recovery, low-rank recovery, electricity plan recommender system.

Nomenclature

n	The number of samples
d	The number of features
X	The sample matrix (kWh) with the size $d \times n$
X^*	The recovered sample matrix (kWh)
E	The noise of the sample matrix (kWh)
Z	The representation matrix of the sample matrix with the size $n \times n$
S	The similarity matrix among customers with the size $n \times n$
Y	Total appliance electricity usages (kWh) with the size $n \times 1$
R	The rating matrix with the size $n \times c$
U_m	The m^{th} customer
U_m^k	The K Nearest Neighbors of the m^{th} customer
Λ	The unknown appliance usage set

Ω	The known appliance usage set
$X_{i,j}$	The i^{th} row and j^{th} column element of matrix X
Y_i	The i^{th} element of vector Y
$X_{i,*}$	The i^{th} row vector of matrix X
$X_{*,i}$	The i^{th} column vector of matrix X
X^t	The t^{th} iteration of X
r	The rank of sample matrix
μ	The penalty parameter
μ_{max}	The maximum penalty parameter
σ_i	Singular value of sample matrix
I	Identity matrix
W	The projection matrix with the size $1\times d$
J	The auxiliary computation matrix with the size $d\times n$
$Y1$, $Y2$	Lagrange multipliers matrices

1 Introduction

An increasing number of factors including intermittent renewable power generation and load consumption have posed a threat to the stability of the power system. These factors cause the fluctuation of the power system and the growing peak value of electricity demand. To deal with the problems, Demand Side Management (DSM) [1-5] is used to regulate the demand of energy consumers. In DSM, with the purpose of shaving peak and filling valley, Pricing Based Demand Response (PBDR) [6-9] is proposed to provide residential customers with various electricity plans, indirectly influencing their energy consumption patterns. For example, if a customer selects a plan with a lower charge in the morning, the customer may shift the use of some appliances from evening to morning.

In a matured electricity market, thousands of electricity plans are listed in the electricity plan interface, which brings challenges to residential customers for making choices among the great number of electricity plans. If a customer chooses an improper electricity plan, to compromise the electricity cost, the customer may have to change the living pattern and sacrifice the living comfort. Faced with this problem, a new technique named Electricity Plan Recommender System (EPRS) is introduced to help residential customers to choose proper electricity plans. In a project named Smart Grid Smart City (SGSC) [10], 200 residential customers are selected to make a comparison between choosing plans with and without EPRS. It shows that aggregated daily load profiles in the two scenarios are similar in shape but slightly different in the lowest and highest values. This project inspires some electricity market platforms, foster them to provide electricity plan recommender service, such as Energy Made Easy, iSelect and Power to Choose [11–13].

The current EPRS methods can be classified into the direct method and indirect method. The direct method is relatively easy to be realized, and the above EPRS models [10-14] belong to this class. These methods directly calculate the residential customers' electricity charges through multiplying their total usages by the unit charge of electricity plans and recommend the electricity plans to make less charges to the residential customers. The main drawback of direct methods is that they lack consideration of the personal needs of customers, because two customers having the same electricity usages may have different living patterns.

In the last decade, the electricity meter can only count for the total appliance electricity usages of customers, so direct methods are the mainstream EPRS methods. Fortunately, with the development of the smart meter, the monitoring of household appliances has become an increasingly attractive research field. Unlike the traditional meters, smart meters and intelligent home devices [14-23] can be utilized to monitor the living patterns of various residential customers, which give the possibility to extract key factors affecting personal living patterns. Based on this technology, indirect methods are introduced to recommend electricity plans based on such factors. Indirect methods are a dual-stage model, consisting of feature formulation stage and recommender stage. In the feature formulation stage, primary data and certain features are set as input and output respectively, and the outputted features are the key factors to represent the living patterns. In the recommender stage, the similarity [24-26] of customers is calculated and the testing personalized electricity plans can be obtained through the calculation based on similarity and training personal electricity plans. The dual-stage framework of EPRS is shown in Fig. 1 below.

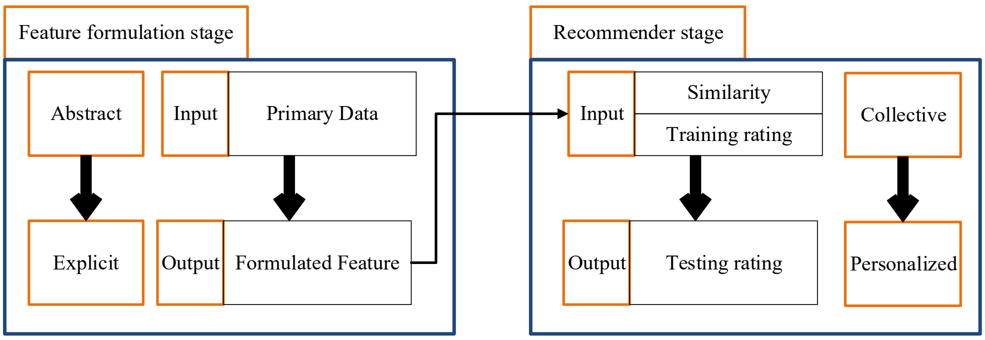


Fig. 1 The dual-stage framework of EPRS.

Similar to the dual-stage of indirect methods, there are two stages as well in these methods. In the feature formulation stage, more explicit features will be obtained. In the recommender stage, personalization will be achieved. The creation will be firstly made in the recommender stage. Reference [27] proposed Cluster-based Recommender System (CB-RS), which set daily electricity usages of different hours as the features and used them to cluster customers. In CB-RS, customers in the same cluster shared the same series of recommended electricity plans. If a new testing feature is inputted into CB-RS, the recommended plans can be computed once the cluster is known. Compared to direct methods, CB-RS proposed clustering methods to recommend personal electricity plans to a new customer based on the cluster, but the disadvantage is that customers in the same cluster are allocated to the same range of electricity plans. The method Social Filtering EPRS (SF-EPRS) in [28] solved this problem by introducing

Collaborative Filtering Recommender System (CFRS) [29] into the recommender stage. In SF-EPRS, the feature formulation stage was similar to that of CB-RS, but in recommender stage, a weighting function was used to compute the recommended electricity plans of a testing customer. With this weighting function, the recommended plans of all testing customers are possible to be different from each other, to satisfy the need for personalization in EPRS. The recommender methods of following papers [30-31] are all based on CFRS.

When the recommender stage is matured, researchers tend to utilize more explicit features in the feature formulation stage. In [30], Collaborative Filtering-based EPRS (CF-EPRS) was proposed. In this model, electricity usages of several selected appliances are transferred into operation duration in the feature formulation stage, and the operation duration is set as features used to compute similarity. Compared with the feature used in CF-EPRS [30], for example, the operation time of washing machine, the feature in CB-RS [27] and SF-EPRS [28], such as the average evening usage or average summer usage, is abstract and implicit. Further progress can be seen in Bayesian Hybrid Collaborative Filtering-based EPRS (BHCF-EPRS) [31]. To avoid the data incompletion, BHCF-EPRS additionally utilizes Probabilistic Matrix Factorization (BPMF) [32] to recover the extracted operation time, which alleviates the sparse problem. Besides, BHCF-EPRS introduces a classification machine to compute the similarity, which makes customers with similar total electricity usages more possible to be set as nearest neighbors. To give the difference of typical indirect methods, Table 1 presents the two stages of these methods.

TABLE 1

Some typical indirect EPRS methods.

Method	Feature extraction stage	Recommender stage
CB-RS [27]	Manual extraction	Cluster
SF-EPRS [28]	Manual extraction	CFRS
CF-EPRS [30]	Turn-on Threshold & Probability Density Function	CFRS
BHCF-EPRS [31]	BPMF	User classification, CFRS

However, for BHCF-EPRS, although BPMF is an applicable way to recovery appliance usages, progress is possible to be made in improving the recovery precision, so we can extract more explicit features. According to [33], a sample matrix is an instinctively low-rank space, which means variables of a sample matrix only depend on a comparably smaller number of factors. The core of matrix recovery is to extract these factors and use them to reconstruct the corrupted data. For example, when predicting the rating of a movie, it is reasonable to assume that the rating may only be determined by a few preferences [34].

In matrix recovery, to exploit low-rank space, two methods are proposed, namely, Matrix Factorization (MF) [34] and Nuclear-Norm Minimization (NNM) [35-37]. The difference between these two methods is the treatment of the rank of the sample matrix. For MF, the rank of the sample matrix and the probability distribution of parameters are set before learning, for example, parameters in BPMF are set to follow Gaussian-Wishart distribution. Instead, NNM [35-37] does not set any prior information into low-rank space and they apply nuclear regularization [38-40] to regulate the sample matrix.

A theorem in [35] shows that NNM can achieve global optimum if the sample matrix is relatively complete, while MF does not have similar convergence property.

In this paper, to get more explicit features, we introduce novel NNM methods to recover appliance usages. According to the difference of recovery principles, we reformulate two classic NNM methods into novel ones, and they are Robust Principal Component Analysis (RPCA) [35] and Low-Rank Representation (LRR) [36], which apply nuclear regularization to learn low-rank data matrix and low-rank representation matrix respectively. Different from the prototypes, the novel ones are combined with electrical recovery instructions, which makes novel methods specialize in recovering appliance usages. Therefore, the new methods are named RPCA with Electrical Instructions (PRCA-EI) and LRR with Electrical Instructions (LRR-EI), respectively, and the new EPRS model is named EPRS with Electrical Instruction-based Recovery (EPRS-EI).

The contributions of this paper are as follows:

(1) We propose EPRS-EI, which is a dual-stage model consisting of feature formulation stage and recommender stage. In feature formulation stage, appliance usages are recovered by PRCA-EI or LRR-EI and set as features, while in recommender stage, CFRS with K-Nearest Neighbors and adjusted similarity is applied to recommend personal electricity plans for customers.

(2) Different from the classical matrix recovery models, electrical recovery instructions are applied in PRCA-EI and LRR-EI, which makes matrix recovery models specialize in recovering appliance usages. The recovery instructions we use are appliance classification and total electricity usage. The appliance classification is utilized to keep the known appliance data unchanged, while the total electricity usage is used to make recovered data constrained.

(3) We also utilize a novel adjusted similarity evaluation in CRFS to computing the testing electricity plans. The total electricity usages are introduced into similarity evaluation for computing living pattern similarity among residential customers. In this case, residential customers with similar total electricity usages are more possible to be set as nearest neighbors.

(4) We provide algorithms to solve our proposed methods, together with the convergence behavior and computational complexity analysis. Finally, the results in a recovery simulation and application simulation of EPRS confirm the effectiveness of our proposed methods in comparison to the state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 describes the proposed matrix recovery methods. In Section 3, the framework of the proposed EPRS is proposed. In Section 4, simulation results are conducted and discussed. Finally, the conclusion and future work are presented in Section 5.

2 Proposed Matrix Recovery Methods

2.1 Previous matrix recovery methods

To show the difference between our methods and other methods, Table 2 provides information on the matrix recovery methods.

From Table 2, it can be seen that methods are classified into two parts, and they are MF and NNM. The objective of matrix recovery methods is to learn the low-rank sample matrix and find the intrinsic information in the sample matrix [35]. For MF methods, such as BPMF, the rank and the probability distribution of parameters are set before machine learning, while for NNM methods, for example, RPCA, the rank is adjusted during machine learning. NNM methods are thought as better methods because they perform better in convergence.

In addition to this, NNM methods can be classified into single subspace recovery (like RPCA), and multiple subspaces recovery (like LRR), and the difference is the unit used to recover the sample matrix. For RPCA, the unit is a certain recovered sample, while for LRR, the unit is a series of samples.

To improve performance, additional information is set as recovery instructions. NSHLRR [37] is the method that introduces graph construction and sparse regularization into LRR. Our proposed models also use appliance classification and total electricity usages as recovery instructions, and these instructions are introduced into both RPCA and LRR.

TABLE 2

Some typical matrix recovery methods.

<table>
<tr><th>Method</th><th>Classification</th><th>Recovery base</th><th>Additional instructions</th></tr>
<tr><td>BPMF [32]</td><td>MF</td><td>Gaussian-Wishart distribution</td><td rowspan="3"></td></tr>
<tr><td>RPCA [35]</td><td rowspan="3">NNM</td><td>Single subspace</td></tr>
<tr><td>LRR [36]</td><td rowspan="2">Multiple subspaces</td></tr>
<tr><td>NSHLRR [37]</td><td>Graph construction & Sparse regularization</td></tr>
</table>

Partridge *et al.* [35] proposed RPCA to learn a low-rank matrix with the same dimensions of the sample matrix. The objective function of RPCA is:

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_1 \quad s.t. \quad X = Z + E \tag{1}$$

where $Z \in \mathbb{R}^{d\times n}$ denotes the low-rank sample matrix, $E \in \mathbb{R}^{d\times n}$ denotes the noise matrix. λ is the trade-off parameter of two terms. In Model (1), the first and second terms are nuclear-norm regularization of recovered matrix and L1-norm regularization of the noise matrix, respectively. The constraint is the formulation of the sample matrix. The L1-norm regularization of matrix E is computed by $\|E\|_1 = \sum_{i=1}^{d}\sum_{j=1}^{n}|E_{i,j}|$, and the nuclear-norm regularization of matrix Z is computed by $\|Z\|_* = \sum_{j=1}^{r}|\sigma_i|$ with r denoting the rank of matrix Z and σ_i denoting the i^{th} singular value of matrix Z. Through the optimization of Model (1), the low-rank sample matrix Z can be learned and viewed as the recovered data of sample matrix X.

The principle of RPCA is that nuclear-norm regularization is applied to extract low-rank bases of the sample matrix, utilizing the bases to reconstruct the recovered samples. Details are shown in Fig. 2. In Fig. 2, X_3 and X_6 are selected as a base to recovery the sample matrix.

Liu *et al.* [36] proposed Low-Rank Representation (LRR) to seek the lowest rank representation among all the candidates that can represent the data samples as linear combinations of the bases from primary data, where the objective function is formulated as:

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_1 \quad s.t. \quad X = XZ + E \tag{2}$$

where $Z \in \mathbb{R}^{n\times n}$ denotes the low-rank representation matrix of the sample matrix. λ is a trade-off parameter of two terms. In Model (2), the first term is the nuclear-norm regularization of the representation matrix and the second term is the L1-norm regularization of the noise matrix. The constraint is the formulation of a sample matrix. After obtaining an optimal solution (Z, E), the original data can be recovered by using XZ (or X-E). Since rank(XZ)$\leqslant$rank(X), XZ is also a low-rank recovery to the corrupted data X.

Different from RPCA, LRR introduces low-rank regularization into the representation matrix, instead of the sample matrix. In this case, the data matrix is first to split into various subspaces, and the recovered sample is linearly reconstructed by the samples belonging to the subspace same as the recovered sample. Details are shown in Fig. 2. In Fig. 2, X_1, X_2 and X_3 are allocated to one subspace, while X_4, X_5 and X_6 are allocated into other subspace.

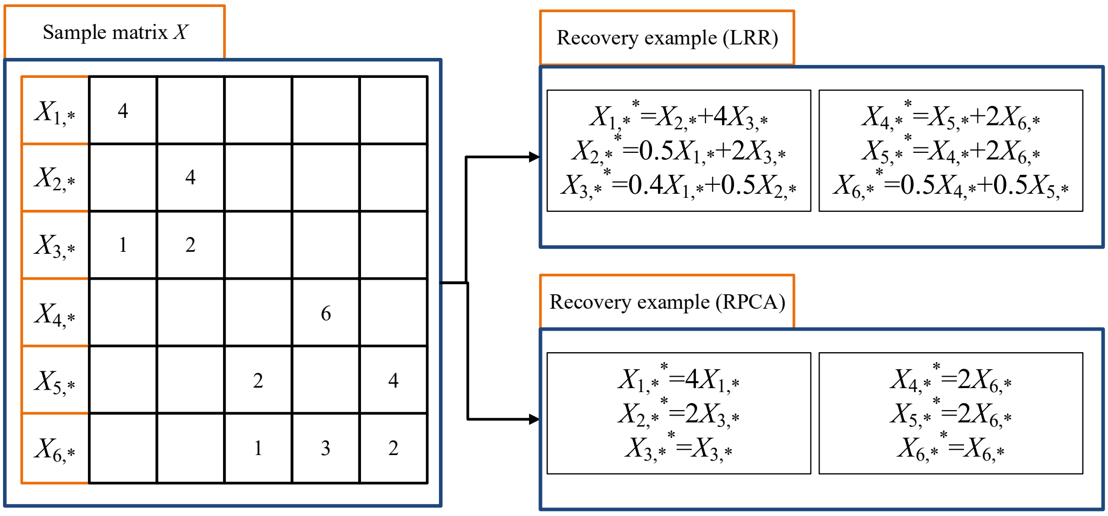


Fig. 2 The recovery example of LRR and RPCA.

Theoretically, this difference of RPCA and LRR lies on the computation flexibility. In [36], it is assumed that the data matrix consists of p low-rank subspaces $\{S_i\}_{1<i<p}$, RPCA can only exploit the sum of these subspaces $\sum_{i=1}^{m} S_i$, while LRR can exploit the union of the subspaces $\bigcup_{i=1}^{m} S_i$. Therefore, LRR can exploit more reconstruction units, having more computation flexibility.

2.2 Matrix Recovery Methods with Electrical Instructions

In this paper, we propose novel methods by introducing electrical recovery instructions into RPCA and LRR according to the specific of the electricity usage, and the novel ones are named RPCA with Electrical Instructions (PRCA-EI) and LRR with Electrical Instruction (LRR-EI).

These electrical recovery instructions are appliance classification and total electricity usage. Firstly, for the appliance classification, there are two appliance categories, that is, the known ones and the unknown ones. The goal is to recover unknown data based on the known appliance data, and the known data is kept unchanged. Secondly, the incompletion can be computed by the difference between total appliance electricity usage and the sum of appliance electricity usages. By introducing total electricity usage into the model, the recovered data can get close to complete total electricity usage.

The objective function of LRR-EI is formulated as:

$$\begin{gathered}\min_{Z,E}\|Z\|_* +\lambda_1 \left\|P_\Omega(E)\right\|_1 +\lambda_2 \|Y\text{-}WXZ\|_F^2 \\ s.t.\ \ X=XZ+E,\ \ P_\Lambda(E)=0\end{gathered} \tag{3}$$

where the third term is the Frobenius-norm regularization of data incompletion error, the second constraint is used to make the noise of measurable appliances zero. λ_1 and λ_2 are trade-off parameters. The projection matrix $W\in\mathbb{R}^{1\times d}$ is a vector that only contains '1', which is used to compute the sum of recovered appliance electricity usage. P_Λ and P_Ω are sets of measurable and measurable appliances, respectively. The Frobenius-norm regularization of matrix X is computed by $\|X\|_F=\sqrt{\sum_{i=1}^{d}\sum_{j=1}^{n}X_{i,j}^2}$.

Since the appliances only contain known and unknown parts, $\Lambda=\overline{\Omega}$ and Model (3) can be transformed into:

$$\begin{gathered}\min_{Z,E}\|Z\|_* +\lambda_1 \left\|P_\Omega(E)\right\|_1 +\lambda_2 \|Y\text{-}WXZ\|_F^2 \\ s.t.\ \ X=XZ+E,\ \ P_{\overline{\Omega}}(E)=0\end{gathered} \tag{4}$$

When $\lambda_2=0$ and the appliance sets are cancelled, LRR-EI degenerates into the primary LRR.

Similarly, the objective function of PRCA-EI is formulated as:

$$\begin{gathered}\min_{Z,E}\|Z\|_* +\lambda_1 \left\|P_\Omega(E)\right\|_1 +\lambda_2 \|Y\text{-}WZ\|_F^2 \\ s.t.\ \ X=Z+E,\ \ P_{\overline{\Omega}}(E)=0\end{gathered} \tag{5}$$

When $\lambda_2=0$ and the appliance sets are cancelled, PRCA-EI degenerates into the primary RPCA.

2.3 Solution

(1) LRR-EI

In this part, the optimization algorithm of LRR-EI is developed.

To make the model unconstrained, the model is reformulated by Augmented Lagrange Method (ALM) [41].

Theorem 1 (ALM) [41]:

With the following objective function:

$$\begin{gathered}\min g(X) \\ \text{s.t.}\ \ h(\mathrm{X})=0\end{gathered} \tag{6}$$

where g: $\mathbb{R}^n\to\mathbb{R}$, h: $\mathbb{R}^n\to\mathbb{R}^m$.

Model (6) can be reformulated to the following Model (7):

$$\begin{gathered}\min L(X,Y1,\mu) \\ \text{s.t.}\ \ L(X,Y1,\mu)=g(X)+tr((Y1)^{\mathrm{T}}h(X))+\frac{\mu}{2}\|h(X)\|_F^2,\ \ \mu>0\end{gathered} \tag{7}$$

where $Y1$ is the Lagrange multiplier and μ is the penalty parameter. The optimum of Model (7) is the same as that of Model (6)

Firstly, by adding an auxiliary matrix J and reformulate Model (3) into:

$$\begin{gathered}\min_{Z,E,J}\|J\|_* +\lambda_1 \left\|P_\Omega(E)\right\|_1 +\lambda_2 \|Y\text{-}WXZ\|_F^2 \\ s.t.\ \ X=XZ+E,\ \ P_{\overline{\Omega}}(E)=0,\ \ Z=J\end{gathered} \tag{8}$$

Secondly, through ALM, Model (8) can be reformulated into the following augmented Lagrangian function:

$$\begin{gathered}\min_{Z,E,J,Y1,Y2}\|J\|_* +\lambda_1 \left\|P_\Omega(E)\right\|_1 +\lambda_2 \|Y\text{-}WXZ\|_F^2 \\ tr((Y1)^{\mathrm{T}}(X\text{-}XZ\text{-}E))+tr((Y2)^{\mathrm{T}}(Z\text{-}J)) \\ +\frac{\mu}{2}(\|X\text{-}XZ\text{-}E\|_F^2+\|Z\text{-}J\|_F^2) \\ \text{s.t.}\ \ P_{\overline{\Omega}}(E)=0\end{gathered} \tag{9}$$

where $Y1$, $Y2$ are Lagrange multipliers, and μ is a positive penalty parameter.

Model (9) is convex and can be optimized by updating Z, E, J, $Y1$ and $Y2$ alternatively. To optimize Model (9), the following lemma should be provided.

Lemma 1 ([42]): Given the following model:

$$\min_W \frac{1}{\alpha}\|W\|_* +\frac{1}{2}\|W\text{-}S\|_F^2 \tag{10}$$

W can be optimized by:

$$W=Udiag\{\max((\sigma_i -\frac{1}{\alpha}),0)\}_{1\leqslant i\leqslant r}V^T \tag{11}$$

where r is the rank of S, σ_i , U and V are obtained by the Singular Value Decomposition (SVD) of S , i.e. $S=Udiag\{\sigma_i\}_{1\leqslant i\leqslant r}V^T$.

Then Z, E, J, $Y1$ and $Y2$ can be subsequently updated. To compute Z when other variables are fixed, Model (9) can be reformulated into:

$$\begin{gathered}f_Z=\lambda_2 \|Y\text{-}WXZ\|_F^2+tr((Y1)^{\mathrm{T}}(X\text{-}XZ\text{-}E))+tr((Y2)^{\mathrm{T}}(Z\text{-}J)) \\ +\frac{\mu}{2}(\|X\text{-}XZ\text{-}E\|_F^2+\|Z\text{-}J\|_F^2)\end{gathered} \tag{12}$$

With complete square formula, Eq. (12) can be reformulated into:

$$f_Z=\lambda_2 \|Y\text{-}WXZ\|_F^2+\frac{\mu}{2}(\left\|X\text{-}XZ\text{-}E+\frac{Y1}{\mu}\right\|_F^2+\left\|Z\text{-}J+\frac{Y2}{\mu}\right\|_F^2) \tag{13}$$

By taking the derivative of f_Z with respect to Z and setting it to zero, we have:

$$\begin{gathered}\frac{\partial f_Z}{\partial Z}=0 \\ \lambda_2 (WX)^T(WX)Z+\mu X^T XZ\text{-}\mu X^T(X\text{-}E+\frac{Y1}{\mu}) \\ +\lambda_2 (WX)^T Y+\mu Z\text{-}\mu(J\text{-}\frac{Y2}{\mu})=0 \\ Z=(\lambda_2 (WX)^T(WX)+\mu X^T X+\mu I)^{-1} \\ (\mu X^T(X\text{-}E+\frac{Y1}{\mu})+\mu(J\text{-}\frac{Y2}{\mu})+\lambda_2 (WX)^T Y)\end{gathered} \tag{14}$$

To compute J when other variables are fixed, Model (9) can be reformulated into:

$$f_J=\|J\|_* +tr((Y2)^{\mathrm{T}}(Z\text{-}J))+\frac{\mu}{2}\|Z\text{-}J\|_F^2 \tag{15}$$

With complete square formula, Eq. (15) can be reformulated into:

$$f_J=\frac{1}{\mu}\|J\|_* +\frac{1}{2}\left\|J\text{-}(Z+\frac{Y2}{\mu})\right\|_F^2 \tag{16}$$

J in Eq. (16) can be solved by Lemma 1.

To compute E when other variables are fixed, Model (9) can be reformulated into:

$$f_E = \lambda_1 \left\|P_\Omega(E)\right\|_1 + tr((Y1)^T (X\text{-}XZ\text{-}E)) + \frac{\mu}{2}\|X\text{-}XZ\text{-}E\|_F^2 \quad s.t. \quad P_{\overline{\Omega}}(E)=0 \tag{17}$$

With complete square formula, Eq. (17) can be reformulated into:

$$f_E = \frac{\lambda_1}{\mu}\left\|P_\Omega(E)\right\|_1 + \frac{1}{2}\left\|X\text{-}XZ\text{-}E+\frac{Y1}{\mu}\right\|_F^2 \quad s.t. \quad P_{\overline{\Omega}}(E)=0 \tag{18}$$

According to [36], Eq. (18) can be solved by $E=P_\Omega(S_{\frac{\lambda}{\mu}}(X\text{-}XZ\text{-}E+\frac{Y1}{\mu}))$, where $S_\tau(x)$ is a shrinkage operator denoted as $S_\tau(x)=\text{sgn}(x)\max(|x|\text{-}\tau,0)$.

Finally, since $Y1$ and $Y2$ are Lagrange multipliers, and μ is a positive penalty parameter, these matrices can be updated by the inexact ALM in [42].

From the above analysis, the overall algorithm for optimizing LRR-EI is described in Algorithm 1 as follows:

Algorithm 1: LRR-EI

---

1) Input: X, Y and trade-off parameter λ.
2) Initialization: Initialize $J^0=0$, $E^0=0$, $Y_1=Y_2=0$, $\mu^0=10^{-6}$, $\mu_{max}=10^6$, $\rho=1.05$ and $t=0$.
3) While not converged do
   - Given other variables, update Z^{t+1} via Eq. (14).
   - Given other variables, update J^{t+1} via Eq. (16).
   - Given other variables, update E^{t+1} via Eq. (18).
   - Update the multipliers $Y1$ and $Y2$ via
   - $Y1^{t+1}=Y1^t+\mu^t(X\text{-}XZ^t)$.
   - $Y2^{t+1}=Y2^t+\mu^t(Z\text{-}J^t)$.
   - Update the parameter μ^{t+1} by $\mu^{t+1}=\min\{\rho\mu^t,\mu_{max}\}$.
   - $t=t+1$.
4) End while
5) Output: The recovered data $X^*=P_\Omega(XZ)$.

---

(2) RPCA-EI

Same as LRR-EI, RPCA-EI can be made unconstrained by ALM. In this case, Model (4) can be reformulated into the following augmented Lagrangian function:

$$\begin{gathered}\min_{Z,E,J,Y1,Y2}\|J\|_* + \lambda_1\left\|P_\Omega(E)\right\|_1 + \lambda_2\|Y\text{-}WZ\|_F^2 \\ tr((Y1)^T(X\text{-}Z\text{-}E)) + tr((Y2)^T(Z\text{-}J)) \\ +\frac{\mu}{2}\left(\left\|X\text{-}Z\text{-}E+\frac{Y1}{\mu}\right\|_F^2 + \left\|Z\text{-}J+\frac{Y2}{\mu}\right\|_F^2\right) \\ \text{s.t.} \quad P_{\overline{\Omega}}(E)=0\end{gathered} \tag{19}$$

where $Y1$, $Y2$ are Lagrange multipliers, and μ is a positive penalty parameter.

Then we can subsequently update Z, E, J, $Y1$ and $Y2$. To compute Z when other variables are fixed, Model (19) can be reformulated into:

$$f_Z = \lambda_2 \|Y\text{-}WZ\|_F^2 + \frac{\mu}{2}\left(\left\| X\text{-}Z\text{-}E + \frac{Y1}{\mu} \right\|_F^2 + \left\| Z\text{-}J + \frac{Y2}{\mu} \right\|_F^2 \right) \tag{20}$$

By taking the derivative of f_Z with respect to Z and set it to zero, we have:

$$\begin{aligned} Z = &(\lambda_2 W^T W + 2\mu I)^{-1} \\ &(\mu(X\text{-}E + \frac{Y1}{\mu} + J\text{-}\frac{Y2}{\mu}) + \lambda_2 W^T Y) \end{aligned} \tag{21}$$

To compute J when other variables are fixed, Model (19) can be reformulated as

$$f_J = \frac{1}{\mu}\|J\|_* + \frac{1}{2}\left\| J\text{-}(Z + \frac{Y2}{\mu}) \right\|_F^2 \tag{22}$$

J in Eq. (22) can be solved with Lemma 1.

To compute E when other variables are fixed, Model (19) can be reformulated as

$$\begin{aligned} f_E = &\frac{\lambda}{\mu}\left\| P_\Omega(E) \right\|_1 + \frac{1}{2}\left\| X\text{-}Z\text{-}E + \frac{Y1}{\mu} \right\|_F^2 \\ &s.t. \quad P_{\overline{\Omega}}(E) = 0 \end{aligned} \tag{23}$$

it can be solved by $E = P_\Omega(S_{\frac{\lambda}{\mu}}(X\text{-}Z\text{-}E + \frac{Y1}{\mu}))$.

From the above analysis, the overall algorithm for optimizing RPCA-EI is described in Algorithm 2 as follows:

Algorithm 2: RPCA-EI

---

1) Input: X, Y and trade-off parameter λ.
2) Initialization: Initialize $J^0 = 0$, $E^0 = 0$, $Y_1 = Y_2 = 0$, $\mu^0 = 10^{-6}$, $\mu_{max} = 10^6$, $\rho = 1.05$ and $t = 0$.
3) While not converged do
   - Given other variables, update Z^{t+1} via Eq. (21).
   - Given other variables, update J^{t+1} via Eq. (22).
   - Given other variables, update E^{t+1} via Eq. (23).
   - Update the multipliers $Y1$ and $Y2$ via
   - $Y1^{t+1} = Y1^t + \mu^t (X\text{-}Z^t)$.
   - $Y2^{t+1} = Y2^t + \mu^t (Z\text{-}J^t)$.
   - Update the parameter μ^{t+1} by $\mu^{t+1} = \min\{\rho\mu^t, \mu_{max}\}$.
   - $t = t + 1$.
4) End while
5) Output: The recovered data $X^* = P_\Omega(Z)$.

2.4 Convergence Analysis and Complexity Analysis

Firstly, the convergence of the alternating optimization algorithm is discussed. For both RPCA-EI and LRR-EI, the whole model is solved by inexact ALM [36], while Z, J and E are optimized by derivation, Singular Value Thresholding (SVT) [38] and sparse function [39]. These algorithms converge and the proof is demonstrated in [36], [38] and [39].

Then, we analyze the computational complexity of LRR-EI (Algorithm 1) as well as RPCA-EI (Algorithm 2), and the notation O is used to represent the time complexity. As for LRR-EI, according to Eq. (14), Eq. (16), and Eq. (18), the time complexities of calculating Z, J and E are $O(2n^2d+n^3)$, $O(nr^2+n^2r+n^3)$, and $O(nd)$ respectively, where r is the rank of Z. Let t be the iteration number of the overall algorithm, we have $O(t(2n^2d+2n^3+nr^2+n^2r))$ as the overall time complexity of Algorithm 1. In the real application, we have $n \gg d$, and therefore, the time complexity of Algorithm 1 can approximately be taken as $O(2tn^3)$.

For RPCA-EI, according to Eq. (21), Eq. (22) and Eq. (23), the time complexities of calculating Z, J and E are $O(d^3)$, $O(dr^2+ndr+n^3)$ and $O(nd)$ respectively, where r is the rank of Z. Let t be the iteration number of the overall algorithm. Thus, we have $O(t(dr^2+ndr+n^3+d^3))$ as the overall time complexity of Algorithm 2. In the real application, we have $n \gg d$, and therefore, the time complexity of Algorithm 2 can approximately be taken as $O(tn^3)$.

3 Proposed Electricity Plan Recommender System

In this part, a detailed analysis of EPRS with Electrical Instruction-based Recovery (EPRS-EI) is given. The recommender system used in this paper is neighborhood-based collaborative filtering in [29], and we introduce total electricity usage into similarity, to propose a novel adjusted similarity. The process of neighborhood-based collaborative filtering can be viewed as a utility function: $U \times T \rightarrow R$ which can generate a mapping from a set user U and set item T to set rating R. In this paper, the set user U stands for the customers, the set item T stands for the electricity plans users choose, and the set rating R stands for the preference of customers to different electricity plans. The recommender process consists of two stages, and they are feature formulation stage and recommender stage. The detail is shown as follows:

3.1 Feature Formulation Stage

In this stage, we need to get features which represent the customers' living patterns. Appliance usages are set as feature data, but it is impossible to utilize the raw appliance usages because the loss of data is inevitable. The incompletion can be found out when the user's total appliance electricity usage fails to match with the sum of the appliance electricity usages.

To recover the corrupted data, both RPCA-EI and LRR-EI are applied to recover the data, and then the recovered data is inputted into the recommender stage.

3.2 Recommender Stage

In this stage, feature data is transformed into a similarity matrix by similarity evaluation and testing rating is calculated by a weighting function based on training rating and similarity matrix.

(1) Similarity Evaluation

In neighborhood-based collaborative filtering, the neighbors of the testing customers need to be searched. Firstly, KNN is utilized to search for each testing customer's neighbors. Through KNN, we can avoid computing the similarity of pair customers, making it efficient to detect training customers having a similar living pattern to that of the testing customers.

Radial Basis Function (RBF) is then used to evaluate the similarity of the neighbors selected by KNN, which is shown as follows:

$$S_{i,j} = e^{-\frac{\left\|X_{*,i}-X_{*,j}\right\|_2^2}{2\rho^2}} \tag{24}$$

where $S \in \mathbb{R}^{n \times n}$ is the similarity matrix indicating the similarity between pair customers, $X \in \mathbb{R}^{d \times n}$ denotes the feature data of customers which represents customers' electricity living patterns, and $S_{i,j}$ indicates the similarity of the customers

U_i and U_j. The larger the element is, the more similar the pair customers are. The L2-norm regularization in Eq. (24) is computed by $\|Y\|_2 = \sqrt{\sum_{i=1}^{n} Y_i^2}$.

To improve the performance, we introduce total electricity usage into similarity evaluation and call the new similarity adjusted similarity. The computation is:

$$S_{i,j} = \frac{1}{1+\left\|Y_i - WX_{*,i}\right\|_2} \times \frac{1}{1+\left\|Y_j - WX_{*,j}\right\|_2} \times e^{-\frac{\left\|X_{*,i}-X_{*,j}\right\|_2^2}{2\rho^2}} \tag{25}$$

where $\frac{1}{1+\left\|Y_i - WX_{*,i}\right\|_2}$ and $\frac{1}{1+\left\|Y_j - WX_{*,j}\right\|_2}$ are the penalty factors computed by total electricity usages. If the error between customer's total appliance electricity usage and the sum of the appliance electricity usages enlarges, the penalty factors will make the similarity small, and vice versa.

(2) Item Recommender

For the testing customer U_m, let U_m^k be the K-Nearest Neighbors of U_m, the rating of U_m can be calculated by the weighting function as shown below:

$$R_{m,i} = \frac{\sum_{n \in U_m^k} S_{m,n} R_{n,i}}{\sum_{n \in U_m^k} S_{m,n}} \tag{26}$$

If we set the yearly charge of electricity plans as the rating, a lower rating indicates more possibilities that customers choose the electricity plan.

The metric of EPRS is precision for a top N recommendation, and it is calculated by:

$$\text{Precision} = \frac{\left|\tilde{I}_m^N \cap I_m^N\right|}{N} \tag{27}$$

where $\tilde{I}_m^N$ and I_m^N are the set of top N possible predictive and real electricity plans customer U_m would like to choose. In our experiment, N is set to be 6.

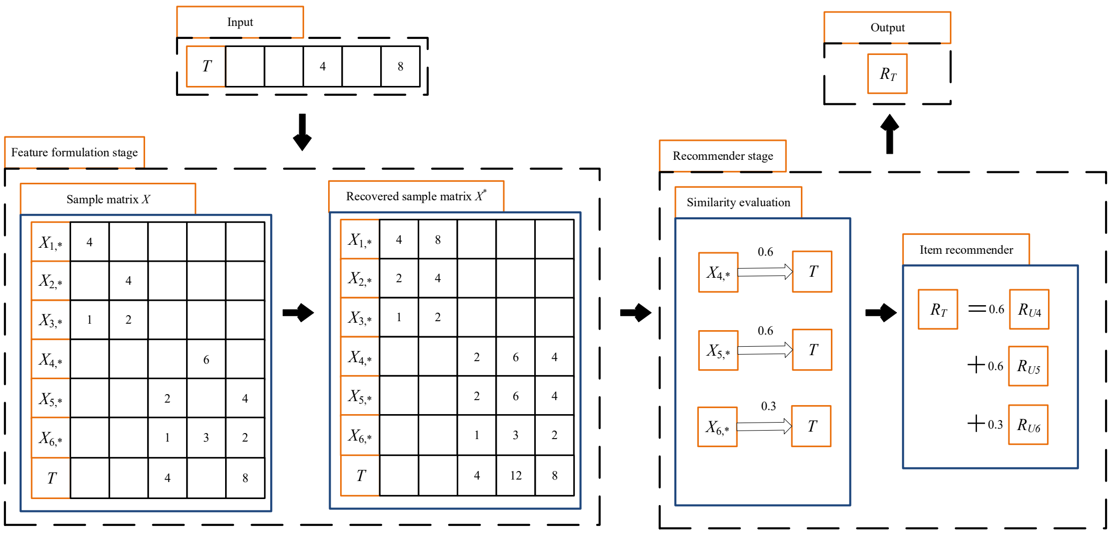


Fig. 3 The framework of EPRS-EI.

3.3 Algorithm and Complexity Analysis

Overall, we have Fig. 3 to show the framework, and the algorithm 3 of EPRS-EI are shown as follows:

Algorithm 3: EPRS-EI

---

1) Input: Appliance electricity usage of training customers X_{tr} and testing customers X_{te}, the yearly charge of electricity plans of training customer is (training ratings) R_{tr}.
2) Recovering X_{tr} and X_{te} by RPCA-EI (or LRR-EI)
3) Detecting neighbors of testing customers via KNN and Eq. (25).
4) Predicting testing customers via Eq. (26).
5) Output: Testing ratings R_{te}.

For EPRS-EI (Algorithm 3), let n_{te} be the number of testing samples and k be the number of their nearest neighbors. We have $O(2tn^3)$ (or $O(tn^3)$), $O(n^2d)$ and $O(n_{te}k)$ as the time complexity of Steps 2-4 for EPRS-EI (LRR-EI) and EPRS-EI (RPCA-EI), and the overall time complexity of EPRS-EI (LRR-EI) and EPRS-EI (RPCA-EI) are $O(2tn^3)$ and $O(tn^3)$ respectively

4 Simulations and discussions

In this part, two types of simulations are conducted. The first type is for recovery, which is utilized to test the recovery capability of the models, while the second simulation is for application, which is utilized to test these models' application potential in EPRS.

4.1 Recovery Simulation

In this simulation, we extract yearly load data from 2014 to 2018 of the three cities

(Austin, Boulder, San Diego), in Dataport [43]. In Dataport, the total appliance electricity usage is the sum of 67 appliance electricity usage. For simplifying the computation, regular electricity usage including wall outlets are cancelled, and only 13 appliances are taken into consideration. Besides, because for a customer, the appliance usage in the weekday and the weekend is different. Therefore, the number of features we set is 26, and the first 13 features represent the usage in the weekdays while the other 13 features represent the usage at the weekends. The selected appliances are shown in Table 3. The data size for 2014 to 2018 data is 622×26, 577×26, 415×26, 364×26, 310×26 respectively.

TABLE 3
The selected appliances.

Air-conditioner	Electric vehicle	Clothes washer	Dishwasher
Dryer	Furnace	Heater	House fan
Microwave	Oven	Range hood	Water heater
Refrigerator			

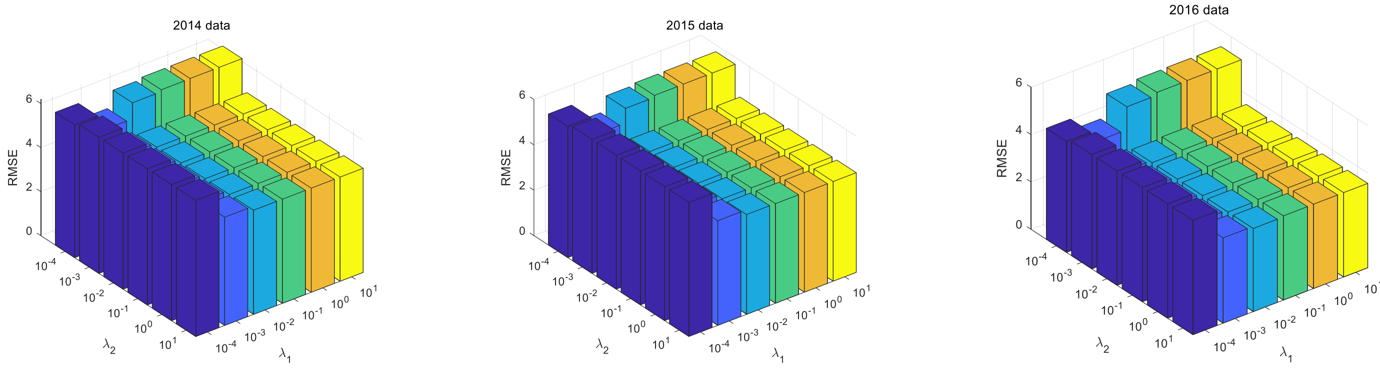
To conduct the recovery, we select a certain number of appliance electricity usage from known appliance electricity usage and allocate those data into unknown data. In this case, the known data is allocated into training data and the unknown one is allocated into testing data. The testing data is recovered based on the remaining data (training data) by matrix recovery.

To avoid bias, 10 runs are conducted in each database by selecting 30%, 40%, 50% and 60% known appliance usage as training data randomly in each yearly data, and 10 cross-validations are adopted to select parameters. After recovering, Root Mean Square Error (RMSE), and RRMSE are computed by:

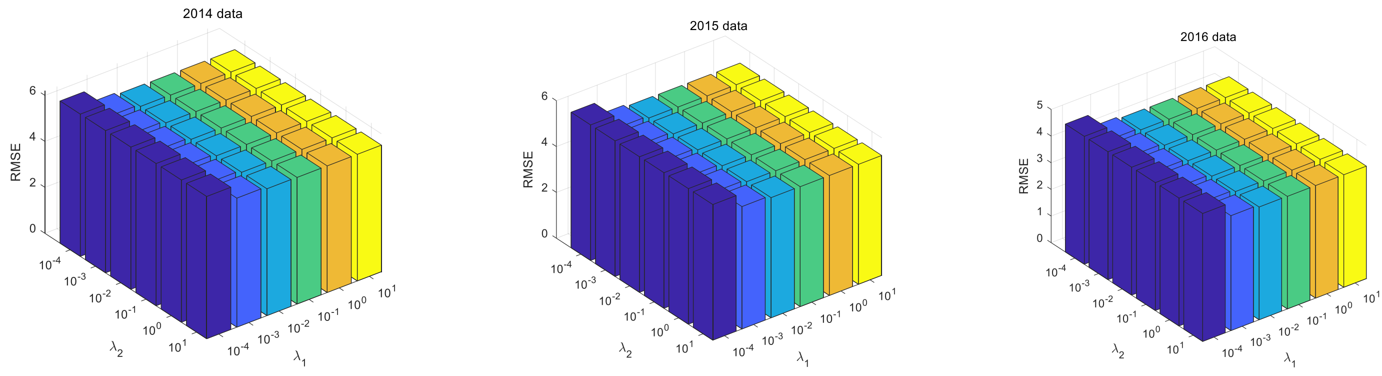
$$\text{RMSE}=\sqrt{\sum_{x_i \in D_{test}} (\tilde{x}_i - x_i)^2} \tag{28}$$

Where $\widetilde{X}$ denotes the mean of the testing data. The lower RMSE indicates better performance.

To test and verify the effectiveness of our proposed method, several competing methods are considered Bayesian Probabilistic Matrix Factorization (BPMF) [32], RPCA [35], LRR [36], NSHLRR [37], RPCA-EI and LRR-EI. All the regulation parameters in the models are chosen from the set $\{10^{-4}, 10^{-3}, \ldots, 10^{1}\}$, the rank parameter in BPMF is set to be 5 times smaller than the number of appliances. The simulation results of RMSE performance are shown in Tables 4-8, and *t*-test result between proposed methods and other methods are shown in Tables 9-13. In *t*-test, “W” means the proposed method performs better. “M” means other approaches perform better. Value in the bracket is the associate p-value. Statistical significance of *t*-test is 5%. The smaller p-value means the higher assurance of the conclusion.



(a)



(b)

Fig. 4 RMSE for different databases with different parameters. (only show 2014-2016's 60% testing data cases) (a) RPCA-EI. (b) LRR-EI.

4.2 Recovery Result Discussions

From Fig. 4, we can see that as the trade-off parameters which have the closest relationship with the objective low-rank matrix may lead to greater change. Comparing with λ_2 , as λ_1 changes, the amplitude of RMSE changes dramatically. This is because λ_1 in both RPCA-RI and LRR-EI, is the parameter which compromises between nuclear-regularization term and other error.

TABLE 4

The RMSE performance for 2014 data.

Training/Total data set / Method	30%	40%	50%	60%	Average
BPMF	13.89 ± 0.22	11.72 ± 0.18	9.83 ± 0.16	8.03 ± 0.09	10.87
RPCA	5.89 ± 0.07	5.37 ± 0.09	4.82 ± 0.09	4.07 ± 0.12	5.03
LRR	5.75 ± 0.07	5.20 ± 0.09	4.61 ± 0.10	3.82 ± 0.13	4.85
NSHLRR	4.47 ± 0.07	5.47 ± 0.08	4.47 ± 0.02	3.17 ± 0.12	4.39
RPCA-EI	5.10 ± 0.07	4.67 ± 0.02	4.21 ± 0.08	3.54 ± 0.13	4.38
LRR-EI	$\mathbf{3.76 \pm 0.08}$	$\mathbf{3.54 \pm 0.08}$	$\mathbf{3.29 \pm 0.09}$	$\mathbf{2.74 \pm 0.14}$	**3.33**

TABLE 5
The RMSE performance for 2015 data.

Method \ Training/Total data set	30%	40%	50%	60%	Average
BPMF	12.66 ± 0.12	10.70 ± 0.13	9.01 ± 0.16	7.39 ± 0.13	9.94
RPCA	5.68 ± 0.11	5.17 ± 0.12	4.61 ± 0.12	3.97 ± 0.11	4.86
LRR	5.55 ± 0.11	5.00 ± 0.12	4.40 ± 0.13	3.72 ± 0.12	4.67
NSHLRR	5.87 ± 0.01	5.17 ± 0.12	4.61 ± 0.12	3.47 ± 0.01	4.78
RPCA-EI	4.90 ± 0.11	4.47 ± 0.12	3.99 ± 0.13	3.44 ± 0.12	4.20
LRR-EI	**3.53 ± 0.16**	**3.28 ± 0.12**	**2.99 ± 0.15**	**2.60 ± 0.14**	**3.10**

TABLE 6
The RMSE performance for 2016 data.

Method \ Training/Total data set	30%	40%	50%	60%	Average
BPMF	12.72 ± 0.21	10.69 ± 0.16	8.98 ± 0.13	7.29 ± 0.16	9.92
RPCA	4.56 ± 0.07	4.16 ± 0.09	3.71 ± 0.11	3.22 ± 0.09	3.91
LRR	4.44 ± 0.07	4.01 ± 0.10	3.52 ± 0.11	3.00 ± 0.09	3.74
NSHLRR	4.56 ± 0.07	3.16 ± 0.09	3.31 ± 0.11	3.12 ± 0.09	3.54
RPCA-EI	3.89 ± 0.06	3.56 ± 0.08	3.19 ± 0.10	2.77 ± 0.08	3.35
LRR-EI	**2.49 ± 0.06**	**2.32 ± 0.05**	**2.11 ± 0.06**	**1.83 ± 0.07**	**2.19**

TABLE 7
The RMSE performance for 2017 data.

Method \ Training/Total data set	30%	40%	50%	60%	Average
BPMF	11.74 ± 0.30	13.87 ± 0.20	11.70 ± 0.23	13.88 ± 0.21	12.80
RPCA	5.28 ± 0.06	5.86 ± 0.11	5.31 ± 0.08	5.81 ± 0.09	5.57
LRR	5.11 ± 0.07	5.73 ± 0.12	5.14 ± 0.08	5.68 ± 0.09	5.41
NSHLRR	4.47 ± 0.09	5.47 ± 0.10	4.47 ± 0.18	5.47 ± 0.19	4.97
RPCA-EIR	4.58 ± 0.07	5.07 ± 0.11	4.60 ± 0.08	5.03 ± 0.09	4.82
LRR-EI	**3.46 ± 0.07**	**3.72 ± 0.11**	**3.48 ± 0.07**	**3.71 ± 0.06**	**3.59**

TABLE 8

The RMSE performance for 2018 data.

Method \ Training/Total data set	30%	40%	50%	60%	Average
BPMF	13.87 ± 0.20	11.74 ± 0.30	9.82 ± 0.15	8.04 ± 0.09	10.87
RPCA	5.86 ± 0.11	5.28 ± 0.06	4.75 ± 0.09	4.10 ± 0.12	5.00
LRR	5.73 ± 0.12	5.11 ± 0.07	4.54 ± 0.09	3.86 ± 0.13	4.81
NSHLRR	5.47 ± 0.01	5.60 ± 0.01	4.47 ± 0.09	3.77 ± 0.01	4.83
RPCA-EIR	5.07 ± 0.11	4.58 ± 0.07	4.13 ± 0.09	3.57 ± 0.13	4.33
LRR-EI	**3.72 ± 0.16**	**3.46 ± 0.07**	**3.17 ± 0.11**	**2.77 ± 0.14**	**3.28**

TABLE 9

t-test result between proposed models and other methods for 2014 data.

Method \ Training/Total data set	RPCA-EI				Method \ Training/Total data set	LRR-EI			
	30%	40%	50%	60%		30%	40%	50%	60%
BPMF	W(.00)	W(.00)	W(.00)	W(.00)	BPMF	W(.00)	W(.00)	W(.00)	W(.00)
RPCA	W(.00)	W(.00)	W(.00)	W(.00)	RPCA	W(.00)	W(.00)	W(.00)	W(.00)
LRR	W(.00)	W(.00)	W(.00)	W(.00)	LRR	W(.00)	W(.00)	W(.00)	W(.00)
NSHLRR	W(.00)	F(.04)	W(.01)	W(.02)	NSHLRR	W(.00)	W(.03)	W(.00)	W(.00)
LRR-EI	M(.00)	M(.00)	M(.01)	M(.00)	RPCA-EI	W(.02)	W(.00)	W(.01)	W(.00)

TABLE 10

t-test result between proposed models and other methods for 2015 data.

Method \ Training/Total data set	RPCA-EI				Method \ Training/Total data set	LRR-EI			
	30%	40%	50%	60%		30%	40%	50%	60%
BPMF	W(.00)	W(.00)	W(.00)	W(.00)	BPMF	W(.00)	W(.00)	W(.00)	W(.00)
RPCA	W(.00)	W(.00)	W(.00)	W(.00)	RPCA	W(.00)	W(.00)	W(.00)	W(.00)
LRR	W(.02)	W(.00)	W(.00)	W(.00)	LRR	W(.00)	W(.00)	W(.00)	W(.00)
NSHLRR	W(.00)	W(.00)	W(.01)	W(.00)	NSHLRR	W(.00)	W(.00)	W(.08)	W(.00)
LRR-EI	M(.01)	M(.00)	M(.00)	M(.04)	RPCA-EI	W(.00)	W(.01)	W(.00)	W(.00)

TABLE 11

t-test result between proposed models and other methods for 2016 data.

Method \ Training/Total data set	RPCA-EI				Method \ Training/Total data set	LRR-EI			
	30%	40%	50%	60%		30%	40%	50%	60%
BPMF	W(.00)	W(.00)	W(.00)	W(.00)	BPMF	W(.00)	W(.00)	W(.00)	W(.00)
RPCA	W(.00)	W(.00)	W(.00)	W(.00)	RPCA	W(.00)	W(.00)	W(.00)	W(.00)
LRR	W(.00)	W(.00)	W(.00)	W(.00)	LRR	W(.00)	W(.00)	W(.00)	W(.00)
NSHLRR	W(.00)	F(.09)	W(.00)	W(.00)	NSHLRR	W(.00)	W(.00)	W(.00)	W(.00)
LRR-EI	M(.00)	M(.00)	M(.02)	M(.00)	RPCA-EI	W(.00)	W(.00)	W(.00)	W(.02)

TABLE 12

t-test result between proposed models and other methods for 2017 data.

Training/Total data set / Method	RPCA-EI				Training/Total data set / Method	LRR-EI			
	30%	40%	50%	60%		30%	40%	50%	60%
BPMF	W(.00)	W(.00)	W(.00)	W(.00)	BPMF	W(.00)	W(.00)	W(.00)	W(.00)
RPCA	W(.00)	W(.00)	W(.00)	W(.00)	RPCA	W(.00)	W(.00)	W(.00)	W(.00)
LRR	W(.00)	W(.00)	W(.00)	W(.00)	LRR	W(.00)	W(.01)	W(.00)	W(.00)
NSHLRR	M(.03)	W(.00)	M(.02)	W(.00)	NSHLRR	W(.01)	W(.00)	W(.00)	W(.00)
LRR-EI	M(.00)	M(.00)	M(.00)	M(.01)	RPCA-EI	W(.00)	W(.00)	W(.00)	W(.03)

TABLE 13

t-test result between proposed models and other methods for 2018 data.

Training/Total data set / Method	RPCA-EI				Training/Total data set / Method	LRR-EI			
	30%	40%	50%	60%		30%	40%	50%	60%
BPMF	W(.00)	W(.00)	W(.00)	W(.00)	BPMF	W(.00)	W(.00)	W(.00)	W(.00)
RPCA	W(.01)	W(.00)	W(.00)	W(.00)	RPCA	W(.00)	W(.00)	W(.00)	W(.00)
LRR	W(.05)	W(.00)	W(.00)	W(.00)	LRR	W(.00)	W(.00)	W(.00)	W(.00)
NSHLRR	W(.01)	W(.00)	W(.00)	W(.00)	NSHLRR	W(.03)	W(.00)	W(.00)	W(.00)
LRR-EI	M(.00)	M(.00)	M(.00)	M(.01)	RPCA-EI	W(.00)	W(.00)	W(.00)	W(.02)

For Tables 4-13, some discussions are given as follows:

- It is known that with the increase of the training samples, RMSE decreases for all methods. This is because as the known information increases, it is easier to recover the appliance electricity usage.
- The performance of BPMF cannot exceed other methods. This is because BPMF belongs to MF method, we need to input the matrix's rank as a parameter. If this rank fails to match well with the data, then it cannot perform well.
- Conversely, other NNM methods can adjust the rank of data matrix automatically in the learning process, and it can be drawn that all methods with instructions, i.e., NSHLRR, RPCA-EI and LRR-EI, outperform their prototypes (RPCA and LRR). The result implies that additional instructions can improve recovery performance, but the additional instructions may cause incorrect recovery results. That is why NSHLRR cannot outperform the prototypes in some databases.
- Fortunately, the performances of RPCA-EI and LRR-EI are better than that of other methods, which indicates that our specific recovery instructions, appliance classification and total electricity usages, are effective in the electronic application.
- To present the effectiveness of proposed methods, $\frac{\text{RMSE}_{pre}\text{-RMSE}_{pro}}{\text{RMSE}_{pre}}\times 100\%$ is proposed to compute the comparisons, where RMSE_{pre} denotes RMSE of the previous methods and RMSE_{pro} denotes that of the proposed methods. Specifically, in Table 4, comparing to NSHLRR ($\text{RMSE}_{\text{NSHLRR}}$=4.39), our method

LRR-EI ($RMSE_{LRR\text{-}EI}$=3.33) gives up to $\frac{4.39\text{-}3.33}{4.39}\times100\%$=24.14% improvement. Similarly, in Table 6, we have $RMSE_{NSHLRR}$=3.54, $RMSE_{LRR\text{-}EI}$=2.19, so LRR-EI gives up to $\frac{3.54\text{-}2.19}{3.54}\times100\%$=38.15% improvement. Improvements can also be seen in other datasets with the range from 24.14% to 38.15%.

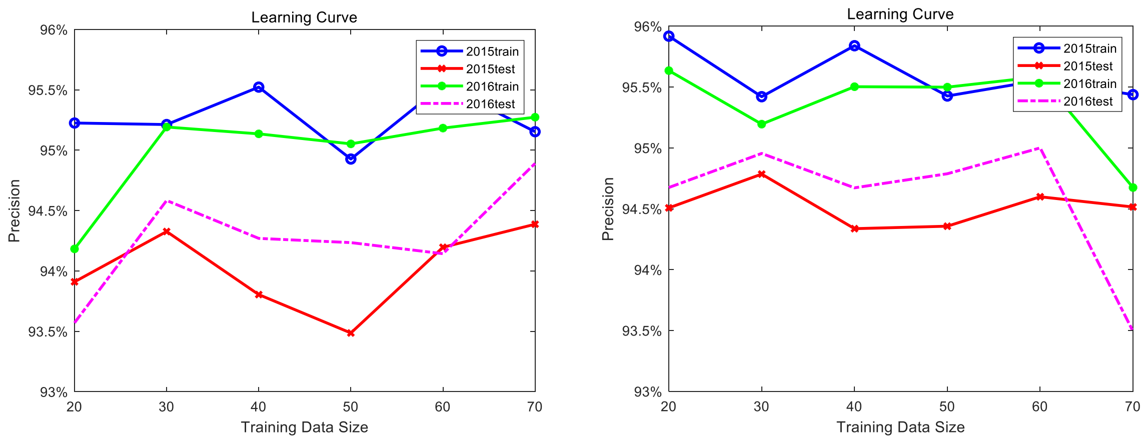
- In Table 4, comparing to LRR-EI's prototype LRR ($RMSE_{LRR}$=4.85), our method LRR-EI ($RMSE_{LRR\text{-}EI}$=3.33) gives up to $\frac{4.85\text{-}3.33}{4.85}\times100\%$=31.22% improvement, while in Table 6, we have $RMSE_{LRR}$=3.74, $RMSE_{LRR\text{-}EI}$=2.19, so LRR-EI gives $\frac{3.74\text{-}2.19}{3.74}\times100\%$=41.50% improvement. Improvements can also be seen in other datasets with the range from 31.22% to 41.50%.
- In Table 4, comparing to RPCA-EI's prototype RPCA ($RMSE_{RPCA}$=5.03), RPCA-EI ($RMSE_{RPCA\text{-}EI}$=4.38) gives up to $\frac{5.03\text{-}4.38}{5.03}\times100\%$=12.99% improvement, while in Table 6, we have $RMSE_{RPCA}$=3.91, $RMSE_{RPCA\text{-}EI}$=3.35, so LRR-EI gives $\frac{3.91\text{-}3.35}{3.91}\times100\%$=14.32% improvement. Improvements can also be seen in other datasets with the range from 12.99% to 14.32%.

4.3 Application Study

In this paper, customers' yearly appliance electricity usage is set as feature data. The rating is the yearly charge ranking of different electricity plans. The electricity plans are collected from Energy Made Easy [10], including 7 time-of-use tariffs and 7 single-rate tariffs.

The rating is calculated from 2015, 2016, 2017, 2018 data, and the feature data used is the year-ahead rating. The data size is 343×26, 314×26, 242×26, 153×26. The number of nearest neighbors is chosen from 2 to 5, N in Eq. (27) is set as 6, 10 runs are conducted in each data by selecting 30%, 40%, 50% and 60% of all the users as training users randomly.

To test and verify the effectiveness of our proposed method, several competing methods are considered including DWR (Data without Recovery), SF-EPRS, BHCF-EPRS, EPRS-EI (RPCA-EI) and EPRS-EI (LRR-EI), where DWR is the EPRS model by setting corrupted data as features. All the regulation parameters in the models are chosen from the set $\{10^{-4},10^{-3},\ldots,10^{1}\}$. Fig. 5 shows the learning curve of EPRS-EI, while Fig. 6 shows precision for these different databases with different parameters. The simulation results of precision performance are shown in Tables 14-17, and *t*-test result between EPRS-EI and other methods are shown in Tables 18-21. In *t*-test, as mentioned before, "W" means EPRS-EI performs better. "M" means other approaches performs better. Also "B" means that EPRS-EI and other approaches cannot outperform each other. Value in the bracket is the associate p-value. Statistical significance of *t*-test is 5% and we have not reported the p-value when the mark is "B". The smaller p-value means the higher assurance of the conclusion.

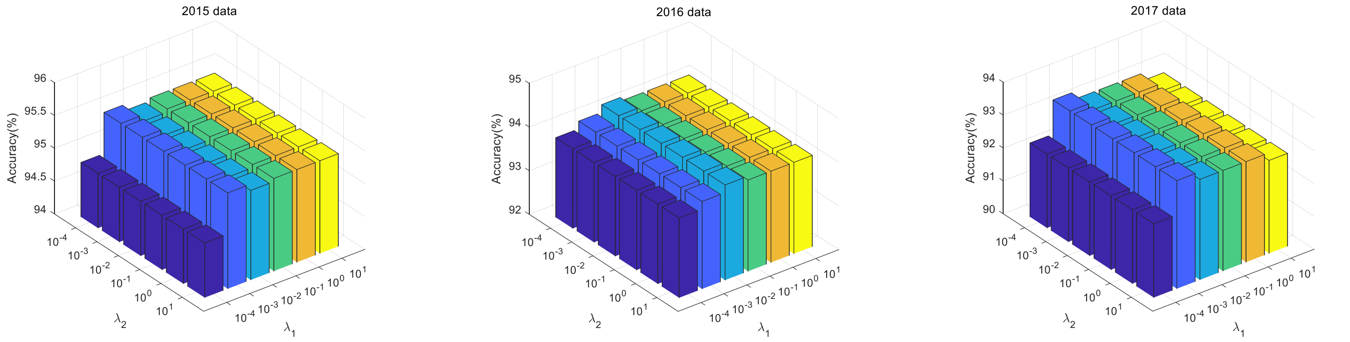


(a) (b)

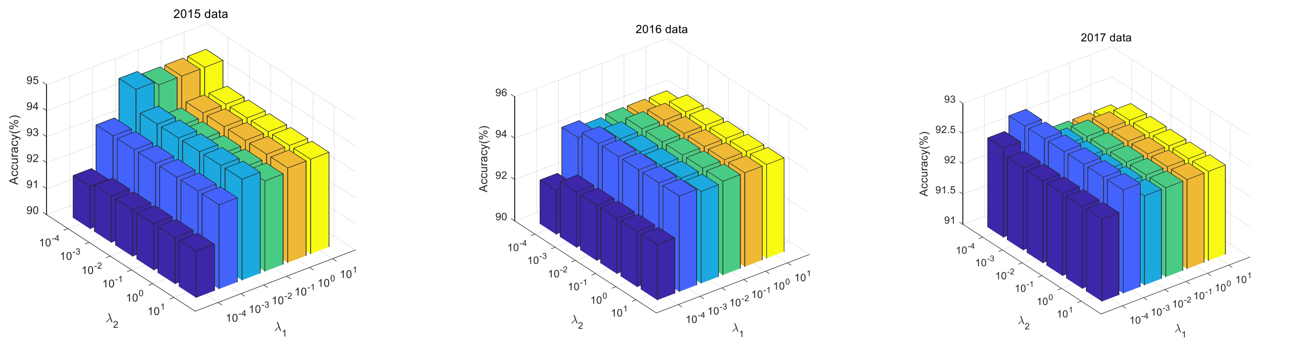
Fig. 5 Learning curve with different training data size (only show 2015-2016's data cases) (a) EPRS-EI (RPCA-EI). (b) EPRS-EI (LRR-EI).

4.4 Application Result Discussions

From Fig. 5, we can see that the simulations in both training and testing data. The horizontal coordinate is training data size, and "20" (to "70") in the coordinate represent the training data takes up 20% (to 70%) of total data and testing data takes up 80 % (to 30%) of total data. When the training data size is enlarged, performance in the testing data cannot exceed performance in the training data, which indicates that our models can avoid overfitting.



(a)



(b)

Fig. 6 Precision for different databases with different parameters (only show 2015-2017's 60% testing data cases) (a) EPRS-EI (RPCA-EI). (b) EPRS-EI (LRR-EI).

From Fig. 6, it can be seen that same as Fig. 4, the trade-off parameter λ_1 , which has the closest relationship with the objective low-rank matrix, is the leading factor to affect the EPRS simulations.

TABLE 14
The precision performance for 2015 data.

Method \ Training/Total data set	30%	40%	50%	60%	Average
DWR	$92.86 \pm 1.19\%$	$92.69 \pm 1.19\%$	$92.03 \pm 0.57\%$	$92.43 \pm 0.69\%$	92.50%
CF-EPRS	$93.24 \pm 0.86\%$	$93.43 \pm 0.60\%$	$93.75 \pm 0.57\%$	$92.80 \pm 0.82\%$	93.30%
BHCF-EPRS	$92.58 \pm 0.99\%$	$93.15 \pm 0.87\%$	$93.71 \pm 0.66\%$	$93.93 \pm 0.89\%$	93.09%
ERPS-EI(RPCA-EI)	$94.78 \pm 0.33\%$	$94.67 \pm 0.44\%$	**$94.36 \pm 0.20\%$**	$94.60 \pm 0.62\%$	**94.60%**
ERPS-EI(LRR-EI)	**$94.83 \pm 0.35\%$**	**$94.77 \pm 0.51\%$**	$94.00 \pm 0.46\%$	**$94.70 \pm 0.56\%$**	94.57%

TABLE 15

The precision performance for 2016 data.

Method \ Training/Total data set	30%	40%	50%	60%	Average
DWR	91.99 ± 1.09%	92.67 ± 1.19%	92.26 ± 1.22%	92.45 ± 1.55%	92.34%
CF-EPRS	93.07 ± 0.73%	93.22 ± 0.74%	93.09 ± 0.69%	92.88 ± 0.78%	93.06%
BHCF-EPRS	93.54 ± 0.50%	93.39 ± 0.69%	93.38 ± 0.72%	93.20 ± 0.82%	93.38%
ERPS-EI(RPCA-EI)	94.95 ± 0.48%	94.67 ± 0.48%	**94.79 ± 0.69%**	**95.00 ± 0.72%**	**94.85%**
ERPS-EI(LRR-EI)	**95.08 ± 0.61%**	**94.77 ± 0.54%**	94.73 ± 0.57%	94.64 ± 0.77%	94.81%

TABLE 16

The precision performance for 2017 data.

Method \ Training/Total data set	30%	40%	50%	60%	Average
DWR	92.66 ± 0.50%	92.36 ± 0.81%	93.06 ± 1.00%	92.99 ± 1.39%	92.77%
CF-EPRS	92.90 ± 0.99%	92.39 ± 0.85%	92.56 ± 0.69%	92.32 ± 0.80%	92.54%
BHCF-EPRS	93.06 ± 0.64%	92.98 ± 0.58%	93.13 ± 0.68%	92.49 ± 0.79%	92.91%
ERPS-EI(RPCA-EI)	93.33 ± 0.54%	**93.86 ± 0.51%**	93.62 ± 0.57%	93.42 ± 0.84%	93.56%
ERPS-EI(LRR-EI)	**93.47 ± 0.28%**	93.77 ± 0.96%	**94.22 ± 0.50%**	**94.69 ± 0.95%**	**93.79%**

TABLE 17

The precision performance for 2018 data.

Method \ Training/Total data set	30%	40%	50%	60%	Average
DWR	92.84 ± 1.14%	91.59 ± 1.40%	91.82 ± 1.09%	92.96 ± 0.73%	92.30%
CF-EPRS	93.41 ± 1.04%	93.28 ± 1.15%	92.59 ± 1.05%	92.90 ± 1.50%	93.04%
BHCF-EPRS	93.54 ± 0.64%	92.92 ± 0.45%	92.87 ± 0.85%	93.11 ± 1.08%	93.11%
ERPS-EI(RPCA-EI)	**93.80 ± 1.24%**	93.73 ± 0.30%	**94.06 ± 0.53%**	**94.04 ± 0.62%**	**93.91%**
ERPS-EI(LRR-EI)	93.74 ± 1.11%	**93.77 ± 0.71%**	94.01 ± 0.59%	94.28 ± 1.02%	93.70%

TABLE 18

t-test result between proposed models and other methods for 2015 data.

Method \ Training/Total data set	ERPS-EI (RPCA-EI)				Method \ Training/Total data set	ERPS-EI (LRR-EI)			
	30%	40%	50%	60%		30%	40%	50%	60%
DWR	W(.00)	W(.03)	W(.00)	W(.00)	DWR	W(.00)	W(.05)	W(.00)	W(.00)
CF-EPRS	W(.00)	W(.00)	W(.01)	W(.00)	CF-EPRS	W(.00)	W(.00)	B(-)	W(.00)
BHCF-EPRS	W(.00)	W(.00)	W(.00)	W(.00)	BHCF-EPRS	W(.00)	W(.00)	W(.00)	W(.00)
ERPS-EI (LRR-EI)	B(-)	B(-)	W(0.03)	B(-)	ERPS-EI (RPCA -EI)	B(-)	B(-)	M(0.03)	B(-)

TABLE 19

t-test result between proposed models and other methods for 2016 data.

Training/Total data set \ Method	ERPS-EI (RPCA-EI)				Training/Total data set \ Method	ERPS-EI (LRR-EI)			
	30%	40%	50%	60%		30%	40%	50%	60%
DWR	W(.04)	W(.00)	W(.00)	W(.03)	DWR	W(.02)	W(.00)	W(.03)	W(.21)
CF-EPRS	W(.00)	W(.00)	W(.00)	W(.00)	CF-EPRS	W(.00)	W(.00)	W(.00)	W(.00)
BHCF-EPRS	W(.00)	W(.00)	W(.00)	W(.00)	BHCF-EPRS	W(.00)	W(.00)	W(.00)	W(.00)
ERPS-EI (LRR-EI)	B(-)	B(-)	B(-)	B(-)	ERPS-EI (RPCA-EI)	B(-)	B(-)	B(-)	B(-)

TABLE 20

t-test result between proposed models and other methods for 2017 data.

Training/Total data set \ Method	ERPS-EI (RPCA-EI)				Training/Total data set \ Method	ERPS-EI (LRR-EI)			
	30%	40%	50%	60%		30%	40%	50%	60%
DWR	W(.17)	W(.09)	W(.05)	W(.29)	DWR	W(.12)	W(.15)	W(.07)	W(.14)
CF-EPRS	B(-)	W(.01)	B(-)	W(.02)	CF-EPRS	B(-)	B(-)	W(.05)	W(.04)
BHCF-EPRS	W(.02)	W(.09)	B(-)	W(.01)	BHCF-EPRS	W(.00)	W(.00)	W(.00)	W(.02)
ERPS-EI (LRR-EI)	B(-)	M(.22)	M(.13)	M(.39)	ERPS-EI (RPCA-EI)	B(-)	W(.22)	W(.13)	W(.39)

TABLE 21

t-test result between proposed models and other methods for 2018 data.

Training/Total data set \ Method	ERPS-EI (RPCA-EI)				Training/Total data set \ Method	ERPS-EI (LRR-EI)			
	30%	40%	50%	60%		30%	40%	50%	60%
DWR	W(.00)	W(.02)	W(.10)	W(.06)	DWR	W(.08)	W(.01)	B(-)	W(.00)
CF-EPRS	B(-)	B(-)	W(.00)	W(.08)	CF-EPRS	B(-)	B(-)	W(.00)	W(.04)
BHCF-EPRS	B(-)	W(.05)	W(.00)	W(.08)	BHCF-EPRS	B(-)	W(.05)	W(.00)	W(.03)
ERPS-EI (LRR-EI)	B(-)	B(-)	W(0.19)	B(-)	ERPS-EI (RPCA-EI)	B(-)	B(-)	M(.19)	B(-)

For Tables 14-21, some discussions are given as follows:

- It is known that in 2017-2018 data (Tables 16-17), with the increase of the training samples, the precision of all methods often rises. This is because as the data size increases, it is much easier to include the training customers having the same living pattern with the testing customer. However, as data size increases in 2015-2016 data (Tables 14-15), we can observe fluctuations. This is because the whole data size of 2015-2016 data is larger than that other data, and ever in the simulations, with the increase in training data, the training customers have no more changes in the diversity of living patterns. Besides, the training customers which influences our result is only 2 to 5 customers, which are set as nearest neighbors, and the number of neighbors is also too small to influence the whole simulations, especially in the large data such as 2015-2016 data.

- In most cases, the performances of DWR fails to have better influence than other methods, and DWR only exceeds other methods when the data size is small, which indicates the effectiveness of the utilization of explicit features and it proves that researchers may be hard to get explicit features when data size is small.
- Also, it can be seen that BHCF-EPRS normally has better performance than CF-EPRS, which proves that BHCF-EPRS can get more explicit features because of the utilization of matrix recovery and user classification. With the user classification technique, even the diversity of customers is scarce in the small data and the nearest neighbors are hard to be detected by similarity evaluation, BHCF-EPRS can also detect nearest neighbors, which strengthen the ability to compute personalized plans.
- Our proposed EPRS-EI still has the best performance, which indicates the effectiveness of the specific recovery instructions, i.e., appliance classification and total electricity usage. All the numbers in Tables 14-17 are computed by Eq. (27). In Table 14, EPRS-EI (RPCA-EI) and EPRS-EI (LRR-EI) have 94.60% and 94.57% of customers correctly recommended respectively. Comparing with BHCF-EPRS (93.09%), their improvements are 94.60%-93.09%=1.62% and 94.57%-93.09%=1.59% respectively. In Table 16, EPRS-EI (RPCA-EI) and EPRS-EI (LRR-EI) do not show such superiority, with 93.56% and 93.79% of customers are correctly recommended respectively. Comparing with BHCF-EPRS (92.91%), their improvements are 93.56%-92.91%=0.70% and 93.79%-92.91%=0.95% respectively. Improvements can also be seen in other datasets with the range from 0.70% to 1.62%.
- Overall, EPRS-EI has the worst performance in Table 16, with 93.56% are correctly recommended, while it has the best performance in Table 15, with 94.85% of customers are correctly recommended.
- It can also be seen that compared to the simulation in recovery simulation, the gap between two proposed methods, i.e. EPRS-EI (RPCA-EI) and EPRS-EI (LRR-EI) is reduced. It indicates that though the data can be well recovered and explicit features can be obtained, the number of nearest neighbors is only 2 to 5 customers, which imposes a limit on the effectiveness of recommender stage. To improve performance, additional techniques such as user classification in BHCF-EPRS can be utilized in EPRS-EI. To invoke user classification, total usages need to be reformulated into regressed targets, which is discrete value, but the total usages in EPRS-EI are set as a continuous value to compute the incompletion between total usages and the sum of appliance usages. Additional preprocessing techniques should be set before user classification is introduced into EPRS-EI.

5 Conclusion and Future Work

Among several electricity tariffs, residential customers can choose a tariff to facilitate DSM. To mitigate the negative effect of corrupted data and recommend personalized electricity plans according to the living patterns, electrical instructions, i.e., appliance classification and total electricity usage, are introduced into improving the performance of EPRS.

Firstly, we propose a novel EPRS model EPRS with Electric Instruction-based Recovery (EPRS-EI) which is a dual-stage model consisting of both feature formulation stage and recommender stage. In the feature formulation stage, novel matrix recovery models with recovery instructions, namely RPCA-EI and LRR-EI are used to obtain more explicit features. These explicit features represent the living patterns of customers. In the recommender stage, KNN and adjusted similarity are utilized to detect nearest neighbors of testing customers, and with the help of the nearest neighbors, testing electricity plans can be computed.

Algorithms are developed to solve the proposed methods. Simulation results on recovery and recommendation confirm the effectiveness of our proposed methods. According to Section 4, LRR-EI recovers up to 24.14%-38.15% of more data compared to NSHLRR, while comparing with BHCF-EPRS, more than 0.70%-1.62% of customers are correctly recommended in EPRS-EI.

Besides, there are two workable improvements for EPRS-EI. The first one is extensive applications to recommend other items. EPRS-EI is a recommender system based on collaborative filtering whose input data consists of feature data and rating. Therefore, if the rating is replaced by other items, this model can be utilized to recommend these items. Two possible recommended items are energy-saving electrical appliances and demand response schedules, and what needs to be done is setting customers' preferences of those appliances or schedules as the rating. The second one is the potential improvement in recommender stage. Since the number of nearest neighbors is limited, if we cannot select the nearest neighbors correctly, the improvement is also limited. The possible selection method is user classification in BHCF-EPRS. However, to invoke user classification, total usages need to be reformulated into regressed targets, additional preprocessing techniques should be set before user classification is introduced into EPRS-EI. In our future work, we also look for the detailed information on electricity plans, usages and charges to accurately compute the saving gained by customers with the help of EPRS.

Acknowledgments

This work is sponsored by the Department of Finance and Education of Guangdong Province 2016 [202]: Key Discipline Construction Program, China; and the Education Department of Guangdong Province: New and Integrated Energy System Theory and Technology Research Group [Project Number 2016KCXTD022].

References

[1] X. Liu, B. Gao, C. Wu and Y. Tang. Demand-side management with household plug-in electric vehicles: a bayesian game-theoretic approach. *IEEE Systems Journal*. 2018; 12(3): 2894-2904. https://doi.org/10.1109/JSYST.2017.2741719.

[2] H. Su, E. Zio, J. Zhang, L. Chi, X. Li and Z. Zhang. A systematic data-driven demand side management method for smart natural gas supply systems. *Energy Conversion and Management*. 2019; 185: 368-383. https://doi.org/10.1016/j.enconman. 2019.01.114.

[3] B. Hayes, I. Melatti, T. Mancini, M. Prodanovic and E. Tronci. Residential demand management using individualized demand aware price policies. *IEEE Transactions on Smart Grid*. 2017; 8(3): 1284-1294. https://doi: 10.1109/TSG.2016.2596790.

[4] N. Liu, X. Yu, C. Wang, C. Li, L. Ma and J. Lei. Energy-sharing model with price-based demand response for microgrids of peer-to-peer prosumers. *IEEE Transactions on Power Systems*. 2017;

32(5): 3569-3583. https://doi.org/10.1109/TPWRS.2017.2649558.

[5] V. Stavrakas and A. Flamos. A modular high-resolution demand-side management model to quantify benefits of demand-flexibility in the residential sector. *Energy Conversion and Management*. 2020; 205: 112339. https://doi.org/10.1016/j.enconman.2019.112339.

[6] S. Yilmaz, J. Chambers and M.K. Patel. Comparison of clustering approaches for domestic electricity load profile characterisation - Implications for demand side management. *Energy*. 2019; 180:665-677: https://doi.org/10.1016/j.energy.2019.05.124.

[7] K. McKenna and A. Keane. Residential load modeling of price-based demand response for network impact studies. *IEEE Transactions on Smart Grid*. 2016; 7(5): 2285-2294: https://doi.org/10.1109/TSG.2015.2437451.

[8] C. S. Lai, F. Xu, M. McCulloch and L. L. Lai. Application of distributed intelligence to industrial demand response. Chen-Ching Liu, Stephen McArthur and Seung-Jae Lee (Editors), Smart Grid Handbook. Piscataway; *IEEE Press & Wiley*; 2016.

[9] F. Y. Xu, T. Zhang, L. L. Lai and H. Zhou. Shifting boundary for price-based residential demand response and applications. *Applied Energy*. 2015; 146: 353–370. https://doi.org/10.1016/j.apenergy.2015.02.001.

[10] Smart Grid, Smart City is a project providing aggregated daily load profile of 200 residential customers with and without EPRS, http://www.environment.gov.au/energy/programs/smartgridsmartcity; 2014. [accessed 4 February 2016].

[11] EnergyMadeEasy is an Australian government website maintained by the Australian Energy Regulator (AER), https://www.energymadeeasy.gov.au/base; 2019 [accessed 4 January 2019].

[12] iSelect is Australia's electricity plan website for comparing plans and saving money, http://www.iselect.com.au/energy/; 2019 [accessed 4 January 2019].

[13] Power to Choose is the official, unbiased, electric choice website of the Public Utility Commission of Texas, http://www.powertochoose.org; 2019 [accessed 4 January 2019].

[14] S. Wang, H. Chen, L. Wu and J. Wang. A novel smart meter data compression method via stacked convolutional sparse auto-encoder, *International Journal of Electrical Power & Energy Systems*. 2020; 118: 105761. https://doi.org/10.1016/j.ijepes.2019.105761.

[15] L. Wen, K. Zhou, S. Yang and L. Li. Compression of smart meter big data: A survey. *Renewable and Sustainable Energy Reviews*. 2018; 91: 59-69. https://doi.org/10.1016/j.rser.2018.03.088.

[16] W. Kong, Z. Y. Dong, D. J. Hill, J. Ma, J. H. Zhao and F. J. Luo. A hierarchical hidden markov model framework for home appliance modeling. *IEEE Transactions on Smart Grid*. 2018; 9(4): 3079–3090. https://doi.org/10.1109/TSG.2016.2626389.

[17] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo and Y. Xu. Improving nonintrusive load monitoring efficiency via a hybrid programming method. *IEEE Transactions on Industrial Informatics*. 2017; 12(6): 2148–2157. https://doi.org/10.1109/TII.2016.2590359.

[18] K.-B. Kim, J. Y. Cho, *et al*. Optimized composite piezoelectric energy harvesting floor tile for smart home energy management. *Energy Conversion and Management*. 2018; 171: 31-37. https://doi.org/10.1016/j.enconman.2018.05.031.

[19] E. Kabalci. A smart monitoring infrastructure design for distributed renewable energy systems. *Energy Conversion and Management.* 2015; 90: 336-346. https://doi.org/10.1016/j.enconman.2014.10.062.

[20] M. M. Eissa. Developing incentive demand response with commercial energy management system

(CEMS) based on diffusion model, smart meters and new communication protocol. *Applied Energy*. 2019; 236: 273-292. https://doi.org/10.1016/j.apenergy.2018.11.083.

[21] B. Yildiz, J. I. Bilbao, J. Dore and A.B. Sproul. Recent advances in the analysis of residential electricity consumption and applications of smart meter data. *Applied Energy*. 2017; 208: 402-427. https://doi.org/10.1016/j.apenergy.2017.10.014.

[22] F. Xu, B. Huang, X. Cun, F. Wang, H. Yuan, L. L. Lai and A. Vaccaro. Classifier economics of semi-intrusive load monitoring. *International Journal on Electrical Power and Energy Systems*. 2018; 103: 224–232. https://doi.org/10.1016/j.ijepes.2018.05.010

[23] J. Zhang, X. Chen, W. W. Y. Ng, C. S. Lai and L. L. Lai. New appliance detection for non-intrusive load monitoring. *IEEE Transactions on Industrial Informatics*. 2019; 15(8): 4819-4849. https://doi.org/10.1109/TII.2019.2916213.

[24] L. Lu, M. Medo, H. Y. Chi, Y. C. Zhang, Z. K. Zhang and T. Zhou. Recommender systems. *Physics Reports*. 2012; 519(1): 1–49. https://doi.org/10.1016/j.physrep.2012.02.006.

[25] X. He, Z. He, J. Song, Z. Liu, Y. Jiang and T. Chua. NAIS: Neural Attentive Item Similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering*. 2018; 30(12): 2354-2366. https://doi.org/10.1109/TKDE.2018.2831682.

[26] H. Kwon and K. Hong. Personalized smart tv program recommender based on collaborative filtering and a novel similarity method. *IEEE Transactions on Consumer Electronics*. 2011; 57(3): 1416–1423. https://doi.org/10.1109/TCE.2011.6018902.

[27] Y. Zhang, K. Meng, D. Xu, M. Lai and Y. Zheng. Recommending electricity plans: A data-driven method. *2016 IEEE International Conference on Smart Grid Communications*. 2016; 668-673. https://doi.org/10.1109/SmartGridComm.2016.7778838.

[28] F. Luo, G. Ranzi, X. Wang, and Z. Y. Dong. Social information filtering based electricity retail plan recommender system for smart grid end users. *IEEE Transactions on Smart Grid*. 2019; 10(1): 95–104. https://doi.org/10.1109/TSG.2017.2732346.

[29] A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang and Y. Li. A survey of matrix completion methods for recommendation systems. *Big Data Mining and Analytics*. 2018; 1(4): 308-323. https://doi.org/10.26599/BDMA.2018.9020008.

[30] Y. Zhang, K. Meng, W. Kong and Z. Y. G. Dong. Collaborative filtering-based electricity plan recommender system. *IEEE Transactions on Industrial Informatics*. 2019; 15(3): 1393-1404. https://doi.org/10.1109/tii.2018.2856842.

[31] Y. Zhang, K. Meng, W. Kong, Z. Y. Dong and F. Qian. Bayesian hybrid collaborative filtering-based residential electricity plan recommender system. *IEEE Transactions on Industrial Informatics*. 2019; 15(8): 4731-4741. https://doi.org/10.1007/978-3-540-24685-5_37.

[32] R. Salakhutdinov and A. Mnih, Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. *Proceedings of the 25th International Conference on Machine Learning*. 2008; 25:880-887. https://doi.org/10.1145/1390156.1390267.

[33] E. J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*. 2010; 98(6): 925–936. https://doi.org/10.1109/jproc.2009.2035722.

[34] A. Mnih and R. R. Salakhutdinov. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*. 2007; 20(2): 1257–1264. https://doi.org/10.3233/IFS-141462.

[35] E. J. Candès, X. Li, Y. Ma and J. Wright. Robust principal component analysis? *Journal of the ACM*. 2009. https://doi.org/10.1145/1970392.1970395.

[36] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank

representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2013; 35(1): 171–184. https://doi.org/10.1109/TPAMI.2012.88.

[37] M. Yin, J. Gao and Z. Lin. Laplacian regularized low-rank representation and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2016; 38(3): 504-517. https://doi.org/10.1109/TPAMI.2015.2462360.

[38] H. Li, N. Chen and L. Li. Error analysis for matrix elastic-net regularization algorithms. *IEEE Transactions on Neural Networks and Learning Systems*. 2012; 23(5): 737-748. https://doi.org/10.1109/tnnls.2012.2188906.

[39] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*. 1996; 58(1): 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[40] B. Recht, M. Fazel and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*. 2010; 52(3): 471-501. https://doi.org/10.1137/070697835.

[41] Z. Lin, M. Chen and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Journal of Structural Biology*. 2013; 181(2): 116-127. https://doi.org/10.1016/j.jsb.2012.10.010.

[42] Y. Zhang. Recent advances in alternating direction methods: practice and theory, http://lsec.cc.ac.cn/~sjom/SJOMInvitedTalks/ProfYZhang.pdf; 2010 [accessed 4 January 2019].

[43] Dataport hosts all the data collected via Pecan Street's water and electricity research, https://dataport.cloud/data; 2019 [accessed 4 January 2019].